
Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [Lajos Foldy](#) on Tue, 11 Dec 2012 21:08:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Cris,

On Tuesday, December 11, 2012 9:14:56 PM UTC+1, Chris Torrence wrote:

> Hi all,

>

> I've been toying around with adding support for the "." operator to HASH. So, for example, you could do the following:

>

> h = HASH()

> h.field1 = "my data" ; adds the key "FIELD1" with data "my data"

> print, h.field1

>

Why? What's wrong with h["field1"]="my data"?

regards,
Lajos

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [David Fanning](#) on Tue, 11 Dec 2012 21:10:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Chris Torrence writes:

> I've been toying around with adding support for the "." operator to HASH.

I think if there is one thing we know about computer programmers, it is that they are not careful readers of documentation. Anything that requires careful reading is guaranteed to be confusing. To be useful, I think software should match expectations, not closely parsed documentation.

I vote for a separate object class.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [Jeremy Bailin](#) on Tue, 11 Dec 2012 22:31:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 12/11/12 2:14 PM, Chris Torrence wrote:

- > Hi all,
- >
- > I've been toying around with adding support for the "." operator to HASH. So, for example, you could do the following:
- >
- > h = HASH()
- > h.field1 = "my data" ; adds the key "FIELD1" with data "my data"
- > print, h.field1
- >
- > So, in essence, this would be a "dynamic" structure.
- >
- > The only problem with this approach is that the keys become all uppercase, because of the case insensitivity of IDL variables. Also, if you add keys during initialization (or using the square brackets), then they would need to be valid IDL variable names, otherwise you couldn't access them using the ".". So no spaces or special characters.
- >
- > I see three possible solutions:
- >
- > 1. Add a keyword to HASH() that forces it into "valid IDL variable" mode. Keys can only be strings. The keys could all be stored as the user provided them, but internally the actual hash would be done with uppercase versions of the keys. The Hash could throw errors if the key wasn't a valid IDL variable name.
- > Advantage: still uses the HASH interface, the case of keys can be preserved & returned to the user
- > Disadvantage: confusing - you could have 2 hashes in your program that behave differently, depending upon a creation keyword that you might not even know was set.
- >
- > 2. Change the HASH behavior, so if a key is a string, then internally it constructs its hash using an "IDL_Validname()" version of the key. Again, we would store the original keys, so they could be returned intact. Numeric keys would be unchanged.
- > Advantage: No weird keyword to have to explain - just a single hash class.
- > Disadvantage: Backwards compatibility issues - could no longer have 2 keys that differed only in their case. Would need to explain that if you want to use "." then you have to be "careful" with your key names.
- >
- > 3. Add a new "DICT" class, that behaves differently than HASH. Keys can only be strings. Keys could still be stored (and returned) with mixed case, errors are thrown if a key isn't a valid IDL

variable name. Internally, the actual hash would be done with uppercase versions of the keys.

- > Advantage: No backwards compatibility issues. Documentation is very clear.
- > Disadvantage: Yet another class.
- >
- > Just to reiterate, in all 3 cases, the original "mixed" case keys would be stored, and could be returned with the hash.Keys() method.
- >
- > Thoughts? Is anyone storing 2 keys that differ only in their case? Are you using HASH, and if so, for what purpose?
- >
- > Thanks!
- > -Chris
- > ExelisVIS
- >

I would strongly argue against 1 or 2. There is zero expectation that a hash will treat its keys as anything other than verbatim, and anything that makes it do so is a bad idea.

3 is fine, but I'm not sure what the point is. (did you see what I did there?)

-Jeremy.

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [markb77](#) on Tue, 11 Dec 2012 23:27:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

I do use the HASH object, it's useful for storing mixed-type metadata, for example. Why change it? I think options 1 and 2 would be very confusing, as the others have commented. Can we have updated widget libraries instead? :)

Mark

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [David Fanning](#) on Tue, 11 Dec 2012 23:55:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

markbates writes:

- > I do use the HASH object, it's useful for storing mixed-type metadata,
- > for example. Why change it? I think options 1 and 2 would be very
- > confusing, as the others have commented. Can we have updated widget
- > libraries instead? :)

Seriously. I had the same thought. It would be interesting to sit in on these meetings where work priorities are established. From the outside, it is difficult to decipher what criteria are used to make these decisions. Certainly not customer clamoring, it would appear, unless I am *completely* out of touch with what is happening in the IDL world.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by chris_torrence@NOSPAM on Wed, 12 Dec 2012 06:05:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Okay, I'll bite.

I have a specific use case for this, where using a "dynamic" structure would be very helpful. Users have also been asking for dynamic, extendable structures for years.

One more data point: It would take less than a day to make this update. Adding a new widget system would take more than a day.

-Chris

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [David Fanning](#) on Wed, 12 Dec 2012 15:31:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chris Torrence writes:

> One more data point: It would take less than a day to make this update. Adding a new widget system would take more than a day.

Good point. But, if you are just looking for stocking

stuffers, here is a problem that I think might take you five minutes to fix (since you do it correctly in other parts of the code) that makes it difficult to write professional looking interfaces for IDL programs. I bring it up because someone asked me about it, again, yesterday. I've been reporting the problem since IDL 6.4, at least.

http://www.idlcoyote.com/widget_tips/nameclip.php

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [markb77](#) on Wed, 12 Dec 2012 16:31:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 12 Dez., 07:05, Chris Torrence <gorth...@gmail.com> wrote:

> Okay, I'll bite.

>

> I have a specific use case for this, where using a "dynamic" structure would be very helpful. Users have also been asking for dynamic, extendable structures for years.

>

> One more data point: It would take less than a day to make this update. Adding a new widget system would take more than a day.

>

> -Chris

hi Chris,

I see what you mean. Regarding the widget stuff, I don't think IDL necessarily needs a whole new widget system, but modernizing the look and features of the existing widgets would be very welcome.

best,
Mark

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [Craig Markwardt](#) on Wed, 12 Dec 2012 19:09:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, December 12, 2012 1:05:08 AM UTC-5, Chris Torrence wrote:

> Okay, I'll bite.

>
>
>
> I have a specific use case for this, where using a "dynamic" structure would be very helpful. Users have also been asking for dynamic, extendable structures for years.

>
>
>
> One more data point: It would take less than a day to make this update. Adding a new widget system would take more than a day.

I would go at it from the other end and make the structure access operator "." overloadable from within the IDL language. Once that key language feature is added, then your particular use case of HASH as the backing store can be wired up with a few lines of IDL code, and you can choose whatever idiosyncracies you wish.

Craig

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by [chris_torrence@NOSPAM](#) on Thu, 13 Dec 2012 03:10:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, December 12, 2012 12:09:34 PM UTC-7, Craig Markwardt wrote:

> On Wednesday, December 12, 2012 1:05:08 AM UTC-5, Chris Torrence wrote:

>
>> Okay, I'll bite.

>
>>
>
>>
>
>>
>
>> I have a specific use case for this, where using a "dynamic" structure would be very helpful. Users have also been asking for dynamic, extendable structures for years.

>
>>
>
>>
>
>>

>
>> One more data point: It would take less than a day to make this update. Adding a new widget system would take more than a day.
>
>
>
> I would go at it from the other end and make the structure access operator "." overloadable from within the IDL language. Once that key language feature is added, then your particular use case of HASH as the backing store can be wired up with a few lines of IDL code, and you can choose whatever idiosyncracies you wish.
>
>
>
> Craig

Hi Craig,

Actually, it is already overloadable. As long as your class inherits from IDL_Object, then using a "." will call SetProperty (if it is on the right-hand-side of the equals), or SetProperty (if it is on the left-hand-side).

And you're right, it was indeed just a couple of lines of code. The trickiness comes in because the IDL parser has already made the "field" name uppercase by the time it reaches the internal C structure code. I really don't want to mess with the IDL parser.

-Chris

Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by chris_torrence@NOSPAM on Fri, 14 Dec 2012 19:33:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, December 12, 2012 8:31:41 AM UTC-7, David Fanning wrote:

>
> Good point. But, if you are just looking for stocking
>
> stuffers, here is a problem that I think might take
>
> you five minutes to fix (since you do it correctly
>
> in other parts of the code) that makes it difficult
>
> to write professional looking interfaces for IDL
>
> programs. I bring it up because someone asked me
>
> about it, again, yesterday. I've been reporting the
>

> problem since IDL 6.4, at least.
>
>
>
> http://www.idlcoyote.com/widget_tips/nameclip.php
>

Well, bad news David. I spent a day looking at this, and it turns out to be a Microsoft Windows vendor bug. Sorry.

-Chris
ExelisVIS
